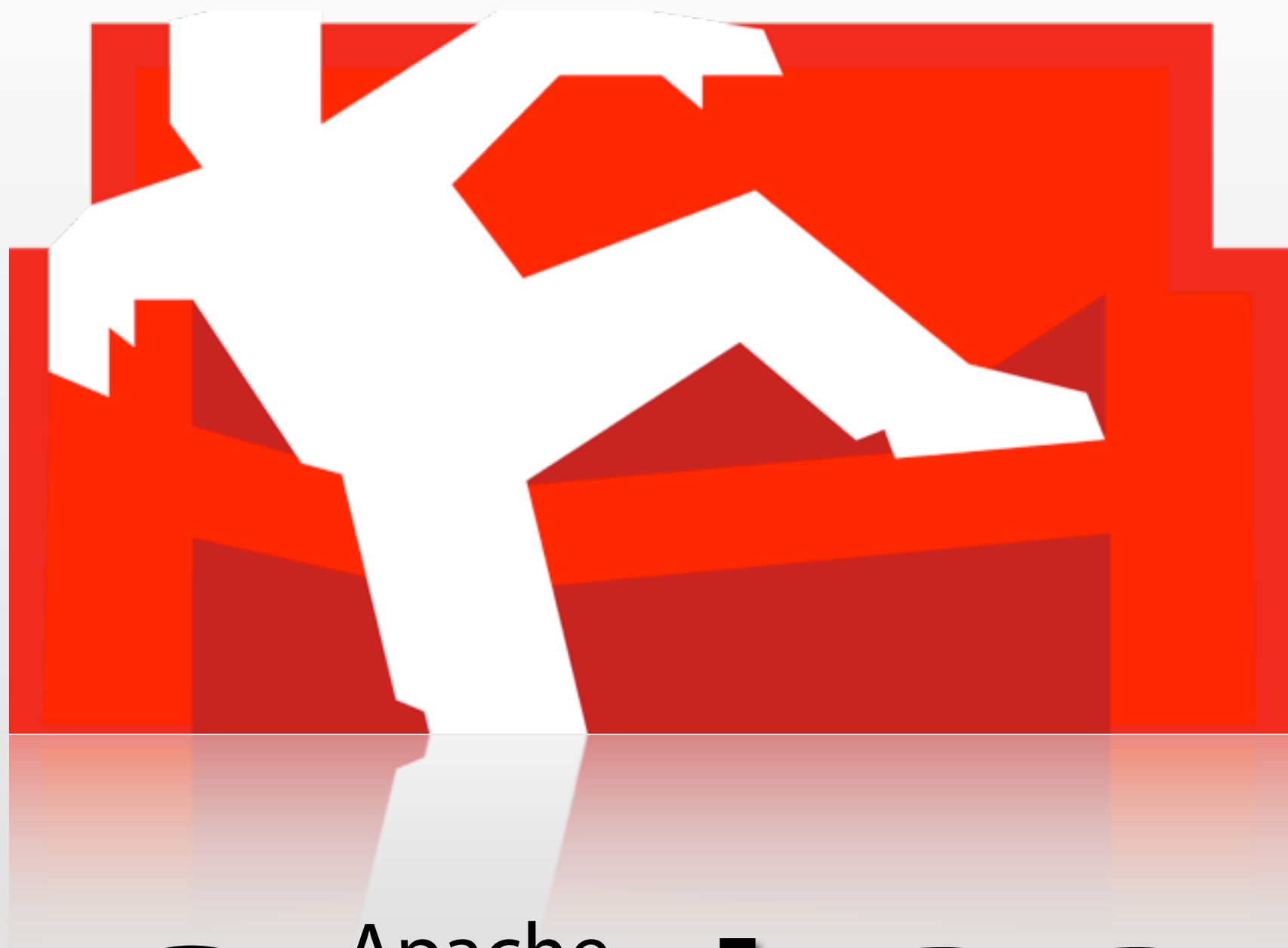




Apache
CouchDB

Relax

Hello

- J Chris Anderson
 - Apache CouchDB committer
 - Web programmer (Rails, Ajax, etc)
- Director, couch.io

Scaling Up



- Hot. Literally.
- You don't have to think.

Scaling Up



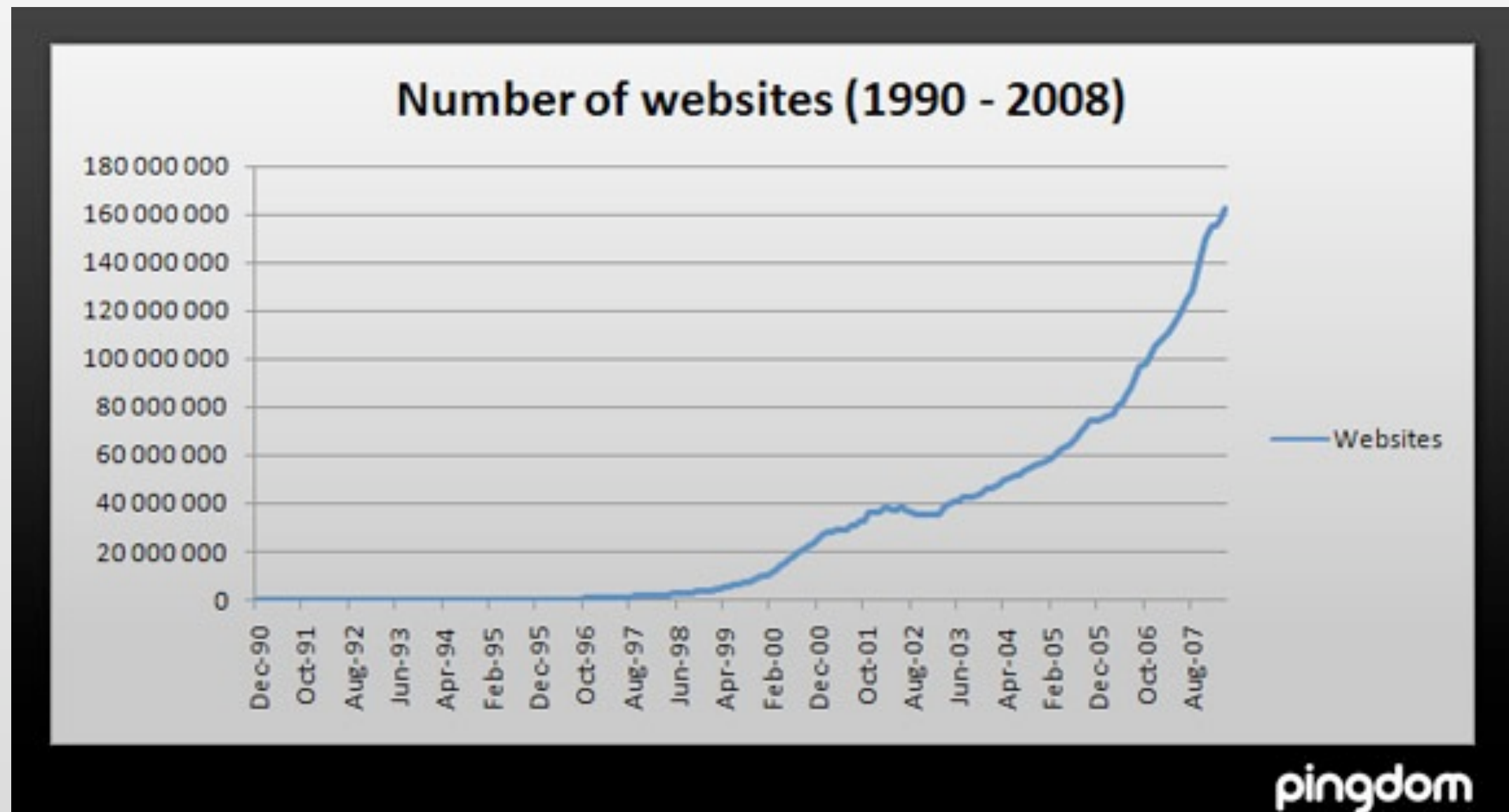
Hot. Little. Easy.

- You don't have to think.

Scale Enforces Constraints

- Distributed key / value stores
 - Context is expensive
- No joins or autoincrement
- Eventual consistency

The Web Scales



Why CouchDB?

Native to the Web

Scale from a cluster to a smartphone

Local data == low latency

Replicated applications - the p2p web

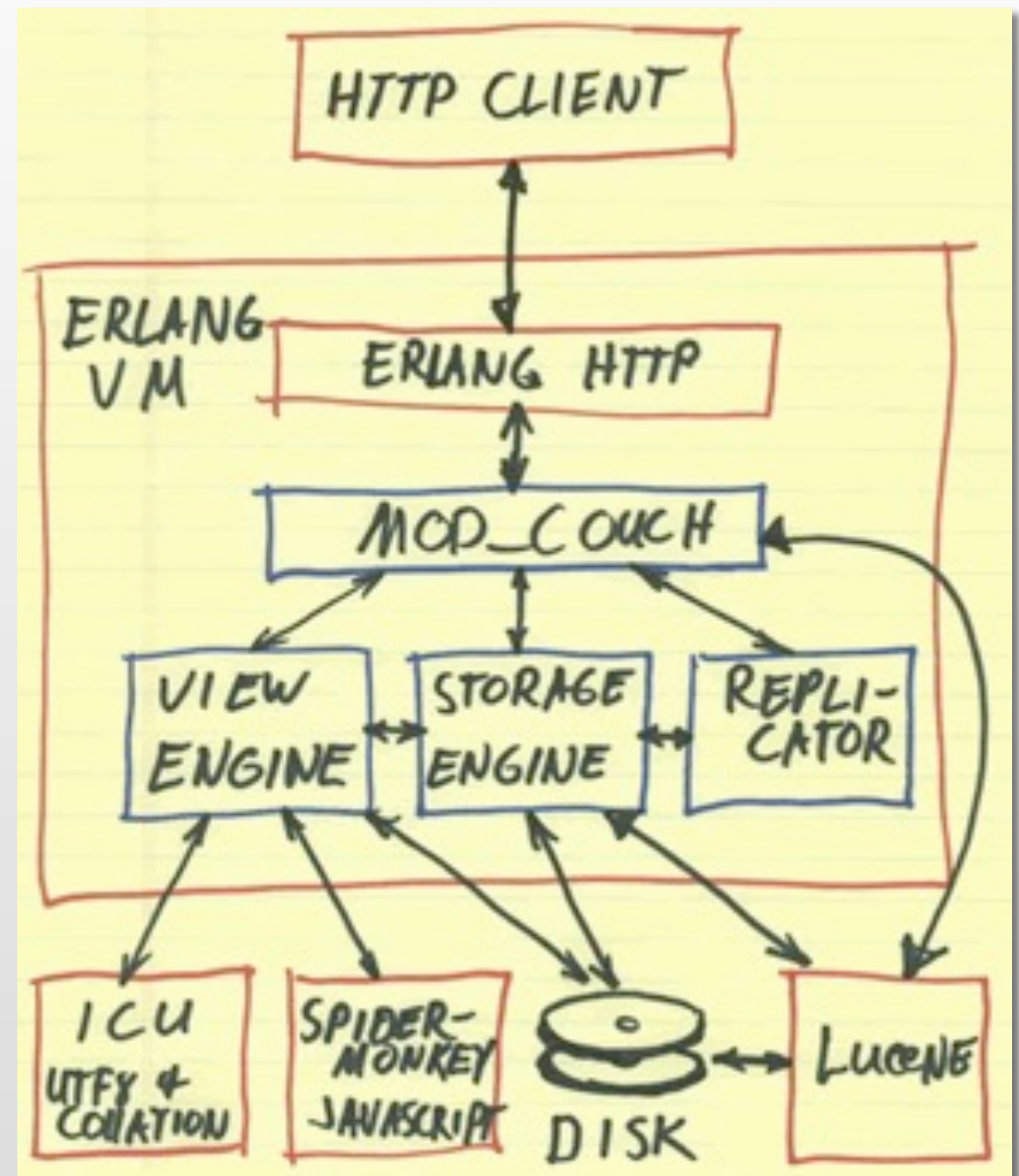
Of the Web

Let me tell you something: Django may be built *for* the Web, but CouchDB is built *of* the Web. I've never seen software that so completely embraces the philosophies behind HTTP. ... this is what the software of the future looks like.

Jacob Kaplan-Moss
October 19, 2007

<http://jacobian.org/writing/of-the-web/>

JSON
HTTP
Local
Robust



JSON Documents

```
{
  "_id": "BCCD12CBB",
  "_rev": "1-AB764C",

  "type": "person",
  "name": "Darth Vader",
  "age": 63,
  "headware":
    ["Helmet", "Sombrero"],
  "dark_side": true
}
```

Schema Free

Self-Contained

JavaScript Map
Reduce Views

Map Reduce Views

Docs

Map

```
{ "user" : "Chris",  
  "points" : 3 }  
{ "user" : "Joe",  
  "points" : 10 }  
{ "user" : "Alice",  
  "points" : 5 }  
{ "user" : "Mary",  
  "points" : 9 }  
{ "user" : "Bob",  
  "points" : 7 }
```

```
function(doc) {  
  if (doc.user && doc.points) {  
    emit(doc.user, doc.points);  
  }  
}
```

```
{ "key" : "Alice", "value" : 5 }  
{ "key" : "Bob", "value" : 7 }  
{ "key" : "Chris", "value" : 3 }  
{ "key" : "Joe", "value" : 10 }  
{ "key" : "Mary", "value" : 9 }
```

Reduce

```
function(keys, values, rereduce) {  
  return sum(values);  
}
```

Alice ... Chris: 15
Everyone: 34

JavaScript Map Reduce

[Overview](#)
[dri](#)

[Create Document ...](#)
[Compact Database](#)
[Delete Database](#)

Select view: tags

View Code

Map Function:

```
function(doc) {
  if(doc.tags && doc.tags.length) {
    for(var idx in doc.tags) {
      emit(doc.tags[idx], 1);
    }
  }
}
```

Reduce Function (optional):

```
function(key, values) {
  return sum(values);
}
```

Run

Language: javascript

=

Revert

Save As...

Save

Key	Value
"#leanintoit"	1
"admin"	1
"Apache"	1
"apachecon"	1
"bookmarklet"	1
"chat"	3
"code"	85
"comet"	2
"Couch"	1
"couchapp"	1

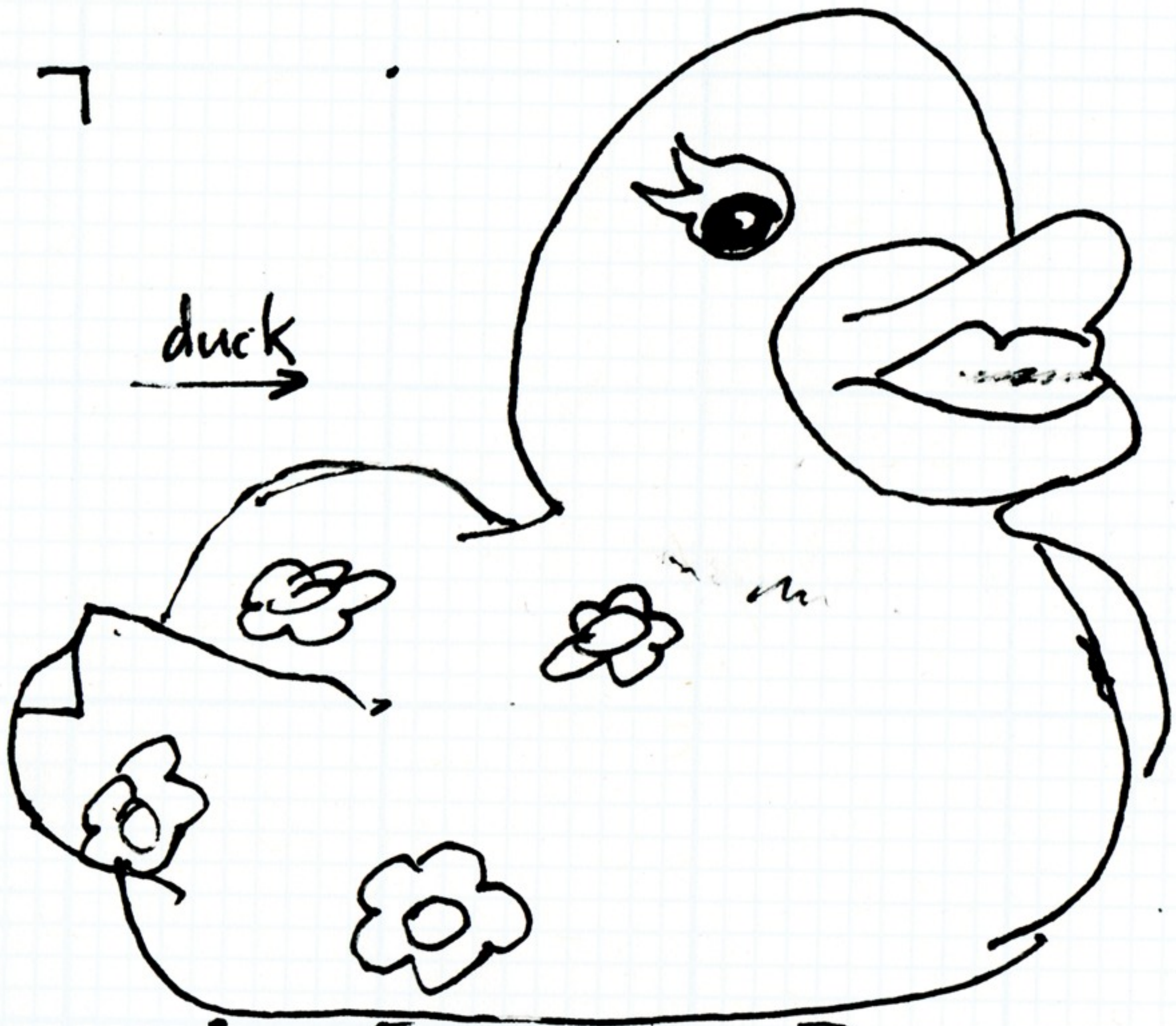
Showing 1-10 of unknown rows

← Previous Page
Rows per page: 10
Next Page →

Note: Views with a reduce function will return a single row of reduced values. If you are looking for the same results that you can see here in Futon, add a `?group=true` parameter to your view query. Futon does this automatically.

7

duck →



REST*ful* HTTP

You already know the API

Use existing HTTP tools

Talk directly to the browser

RESTful HTTP

CRUD

Create

HTTP PUT /db/mydocid

Read

HTTP GET /db/mydocid

Update

HTTP PUT /db/mydocid

Delete

HTTP DELETE /db/mydocid

Partitioned Cluster w/ CouchDB-Lounge



Consistent Hashing

HTTP proxy

Nginx / Twisted Python

Many machines == 1 Couch

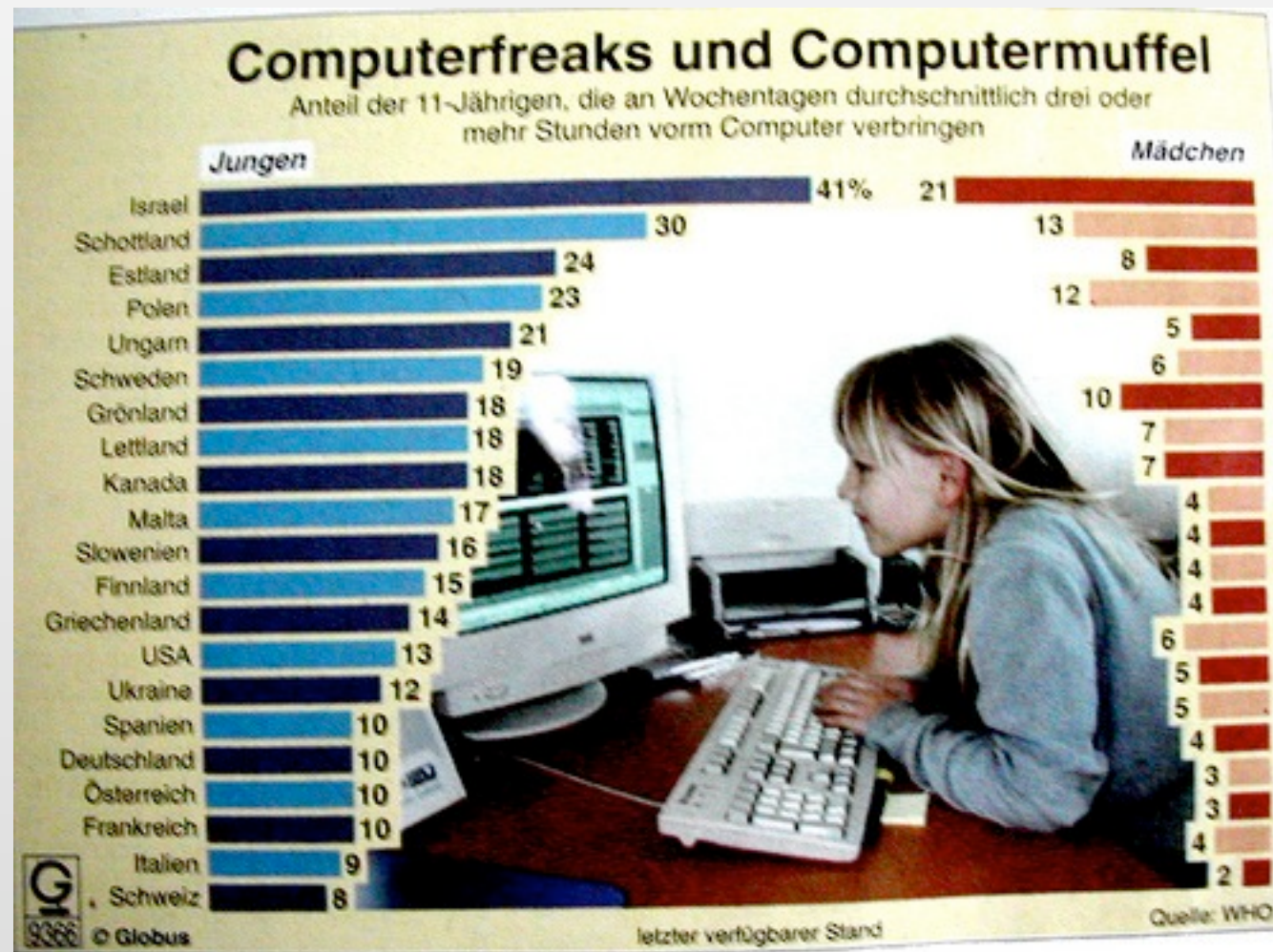
Local Data

Low latency (fast)

Peer-based incremental replication

View source --> Open source

Gives Control to Users



CC-BY-SA <http://www.flickr.com/photos/kelleys/492253912/>

Robust Storage

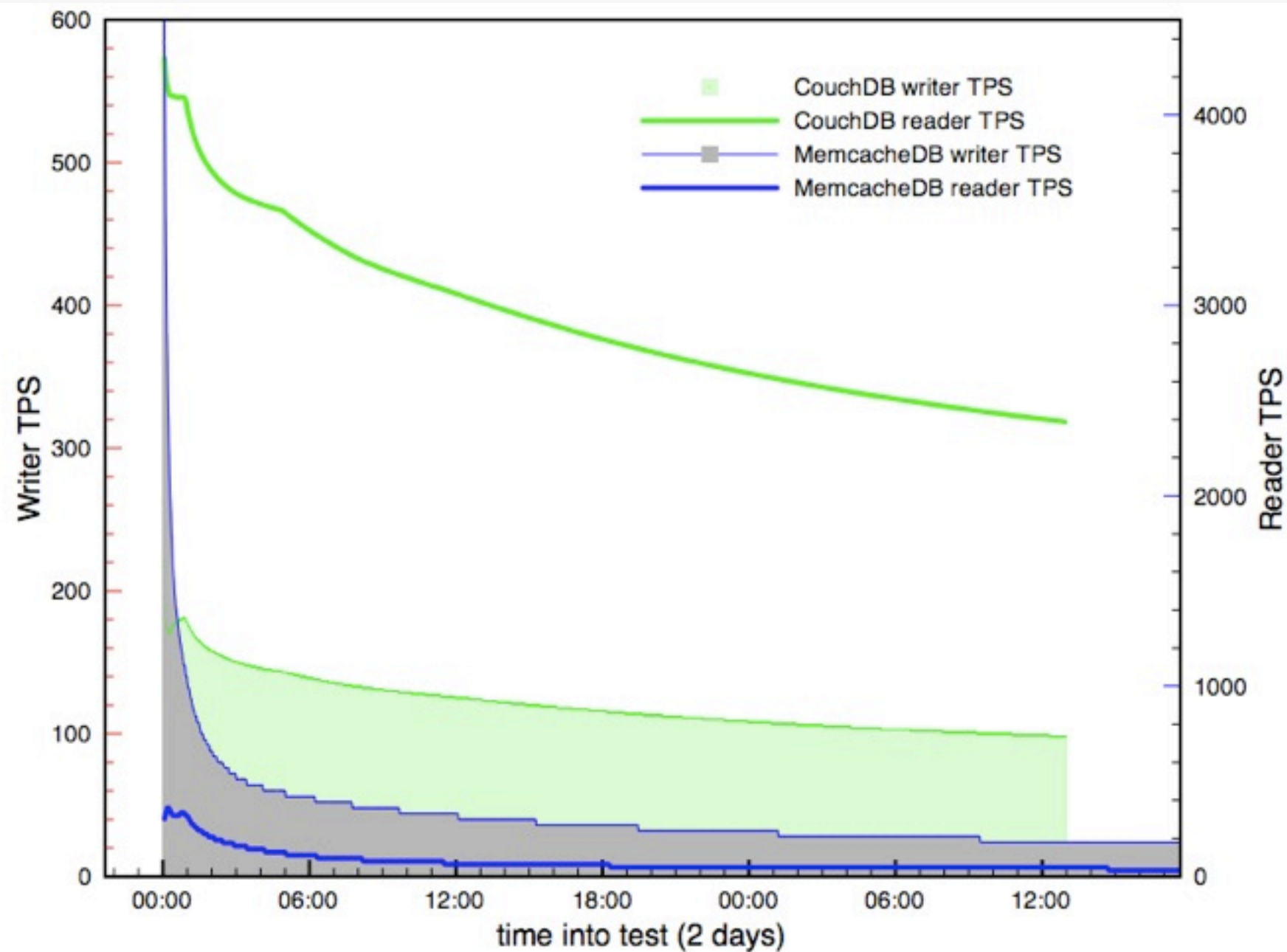
Append Only
File Structure

Designed to Crash

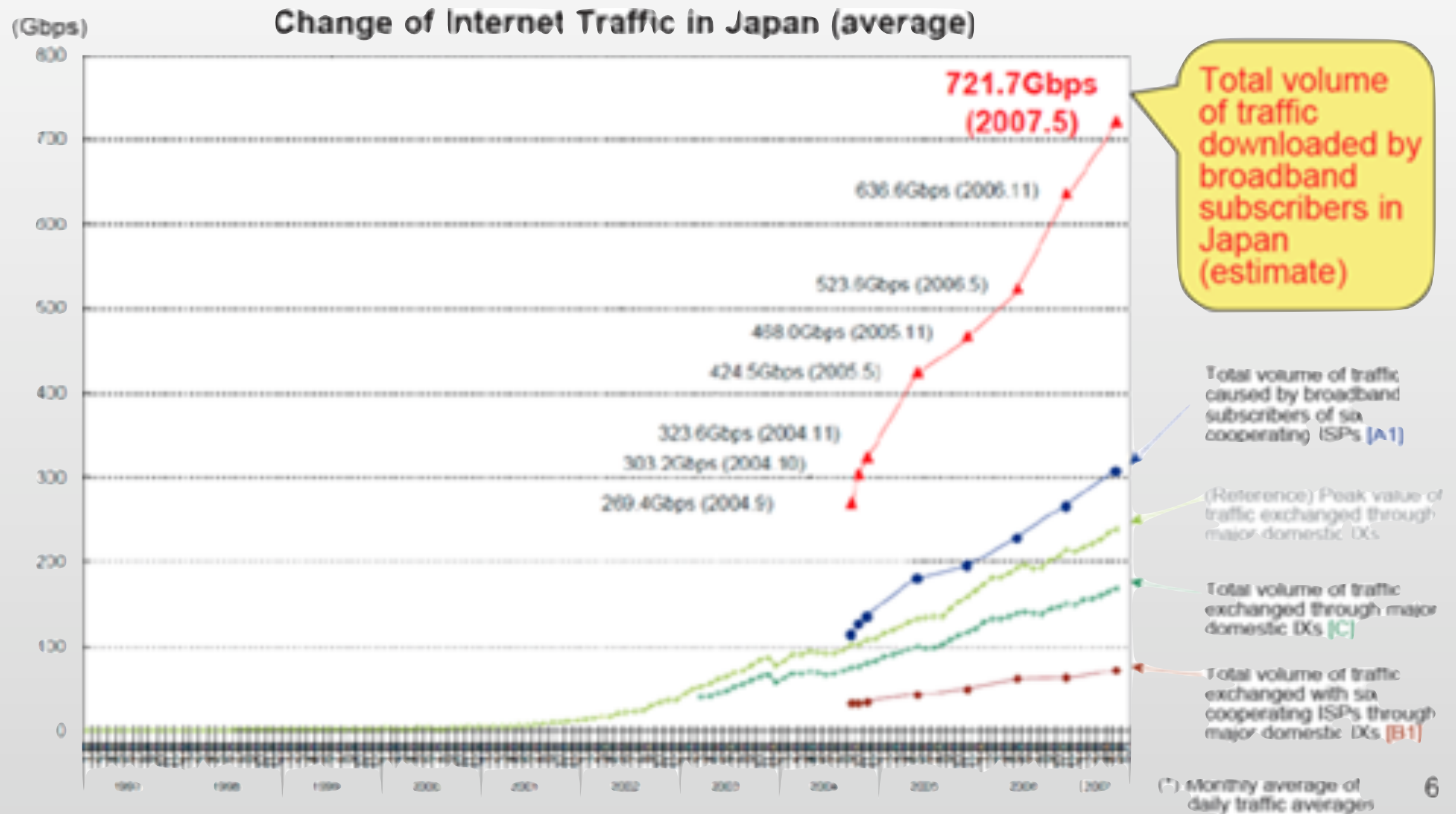
Instant-On

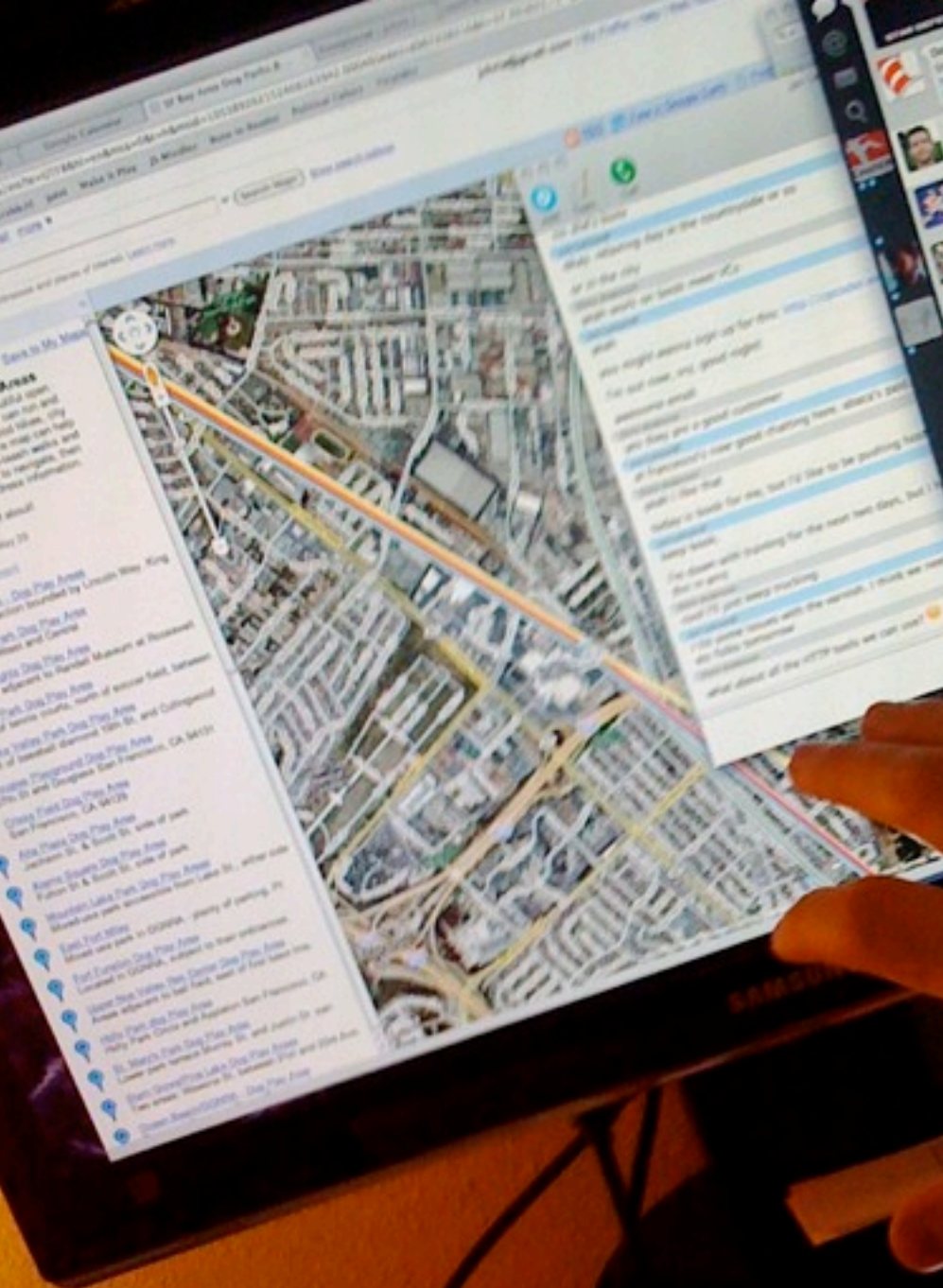


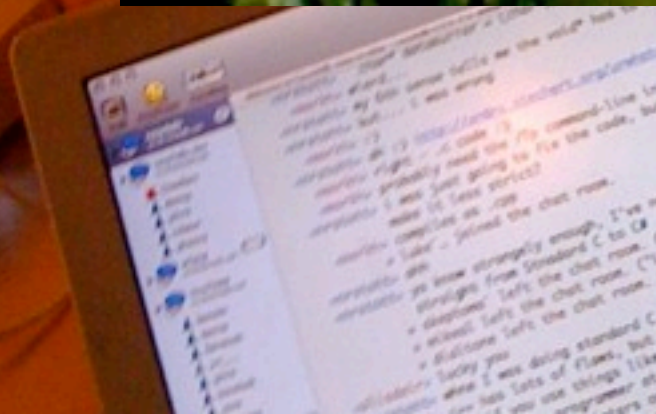
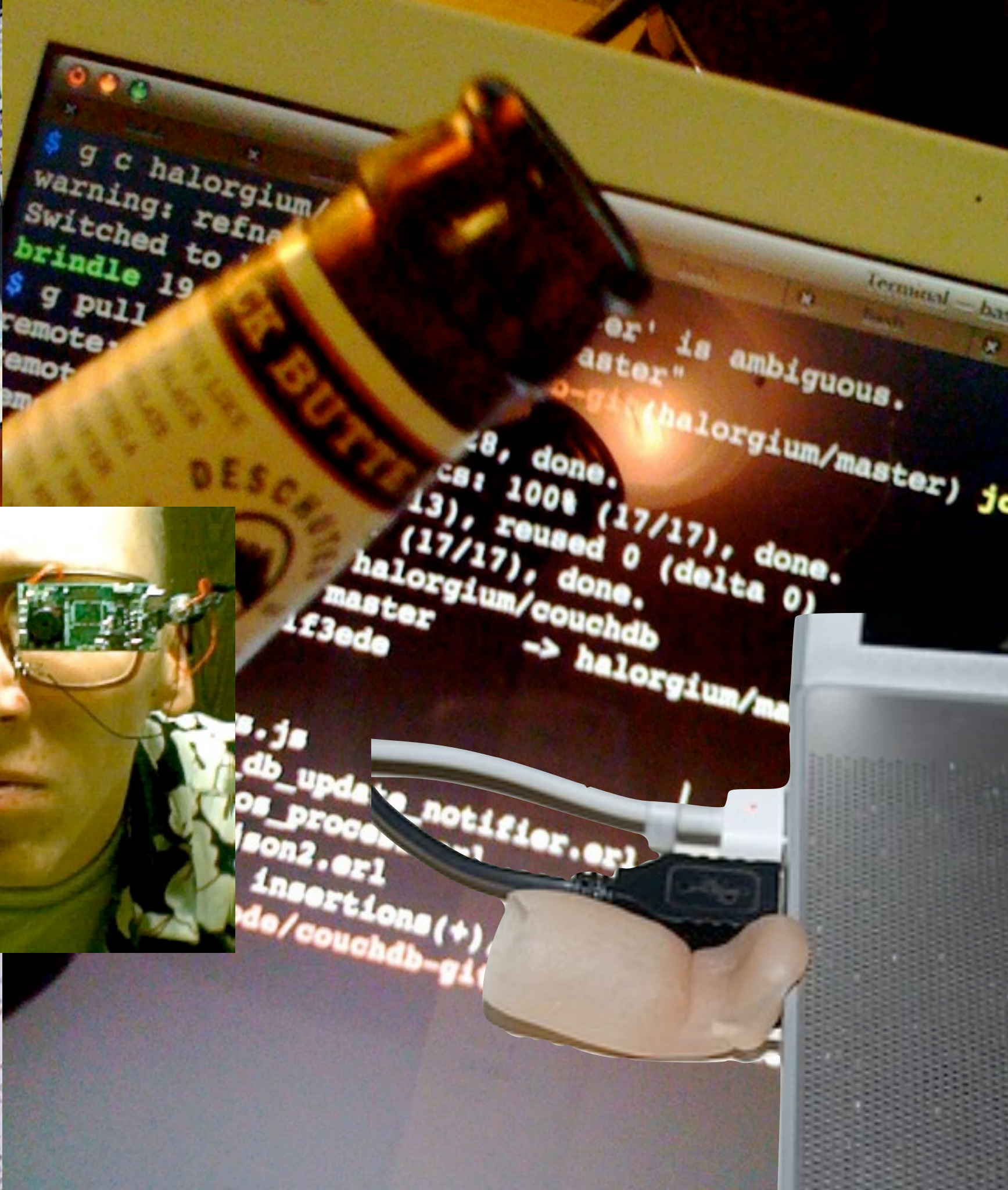
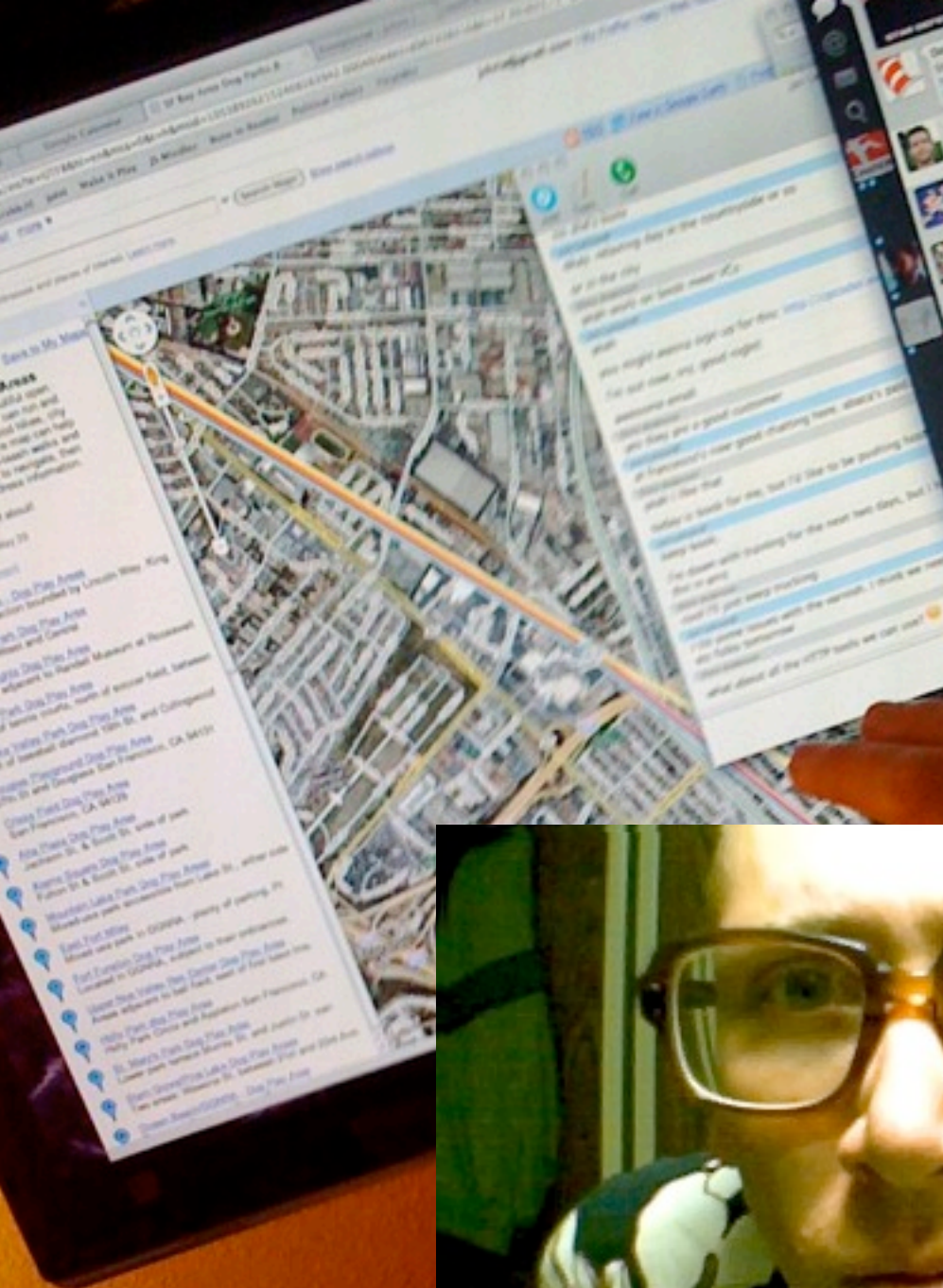
Happy IO Patterns

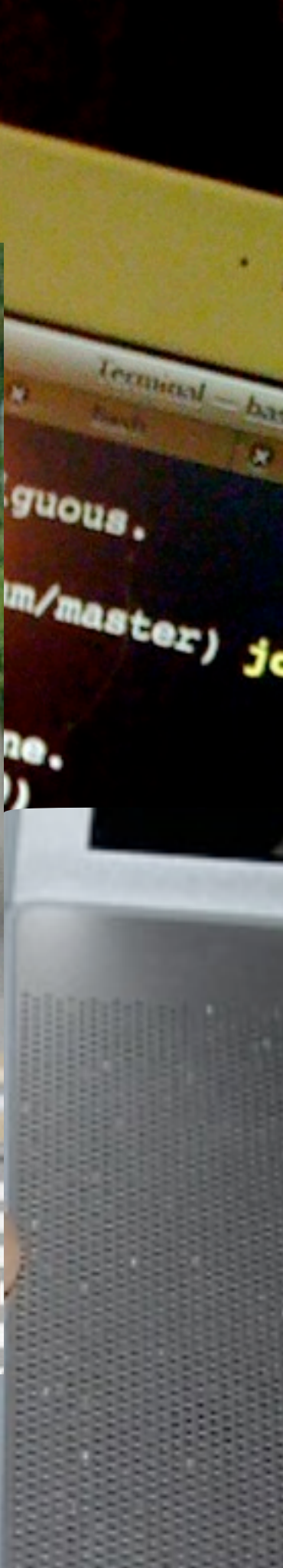
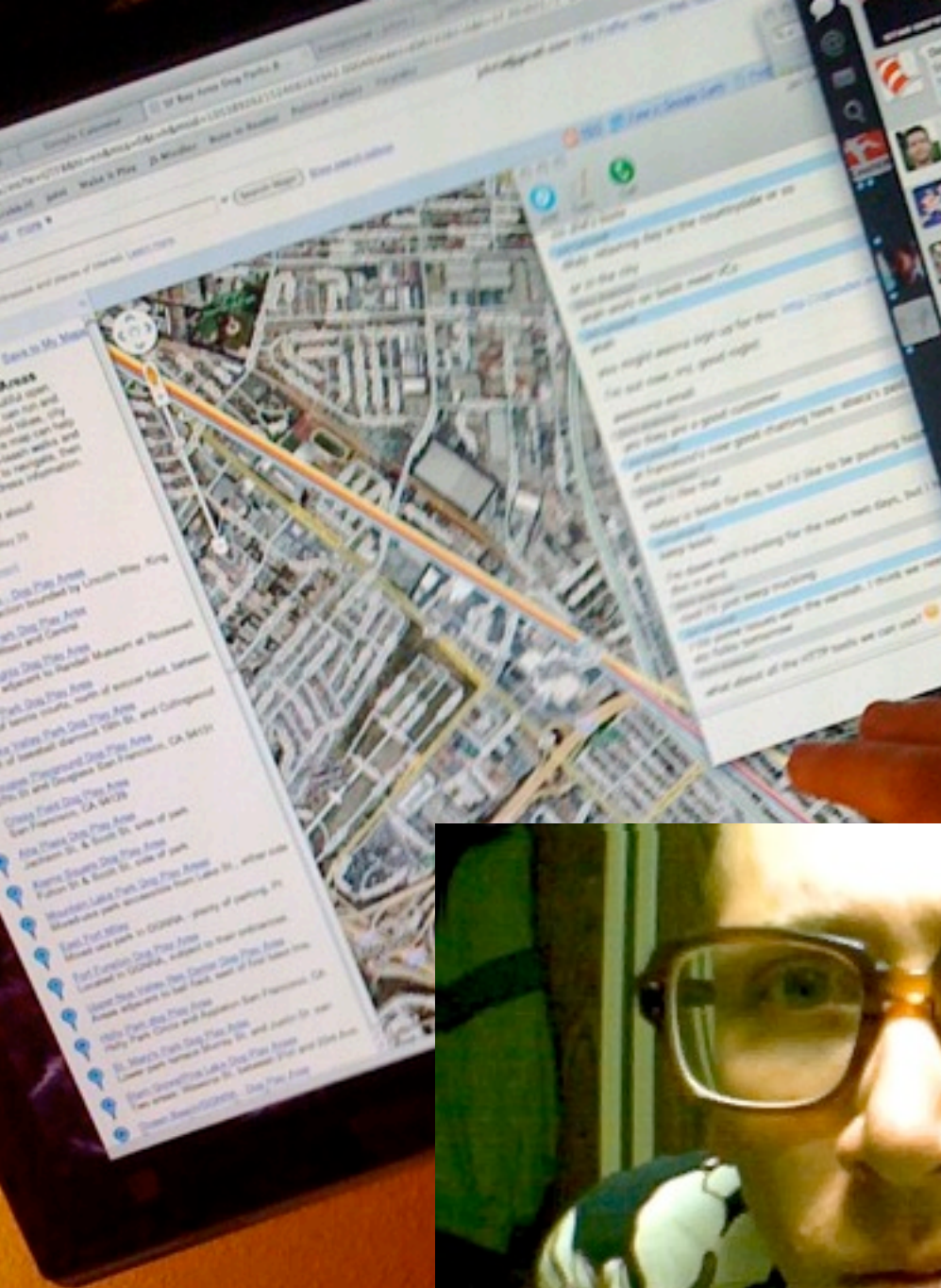


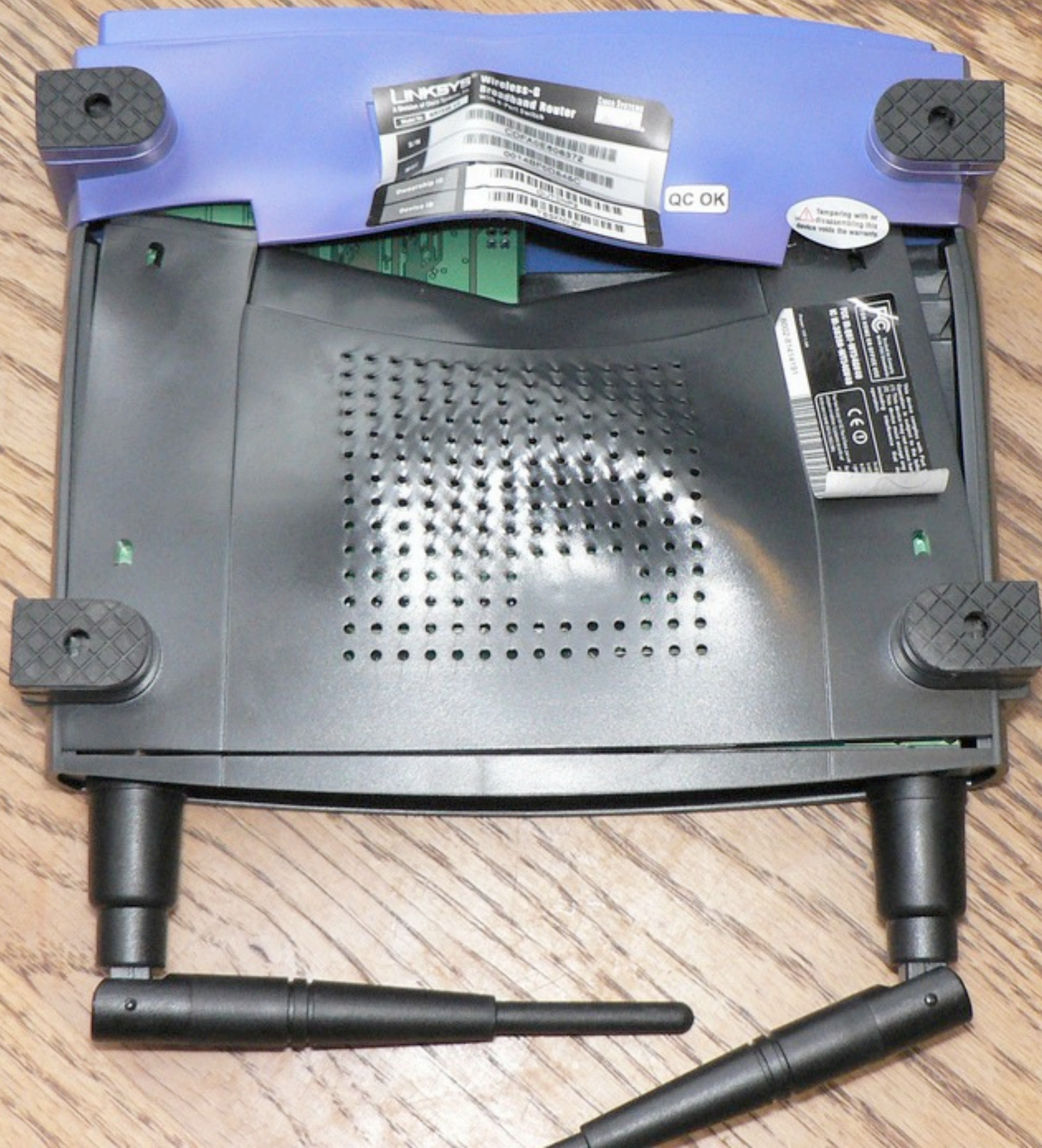
Bandwidth Explosion











“no bars”



FREMONT
5 CARS - BOARD CENTER
PLATFORM 2

1859

1808

Latency Sucks

$$rt^b = rt^a + \sum_{i \in \text{Links}} (l_i^b - l_i^a) \cdot \vec{\nabla} \bar{rt}(l_i)$$

“Ground Computing”

@jhuggins

Scaling Down

4 MB RAM



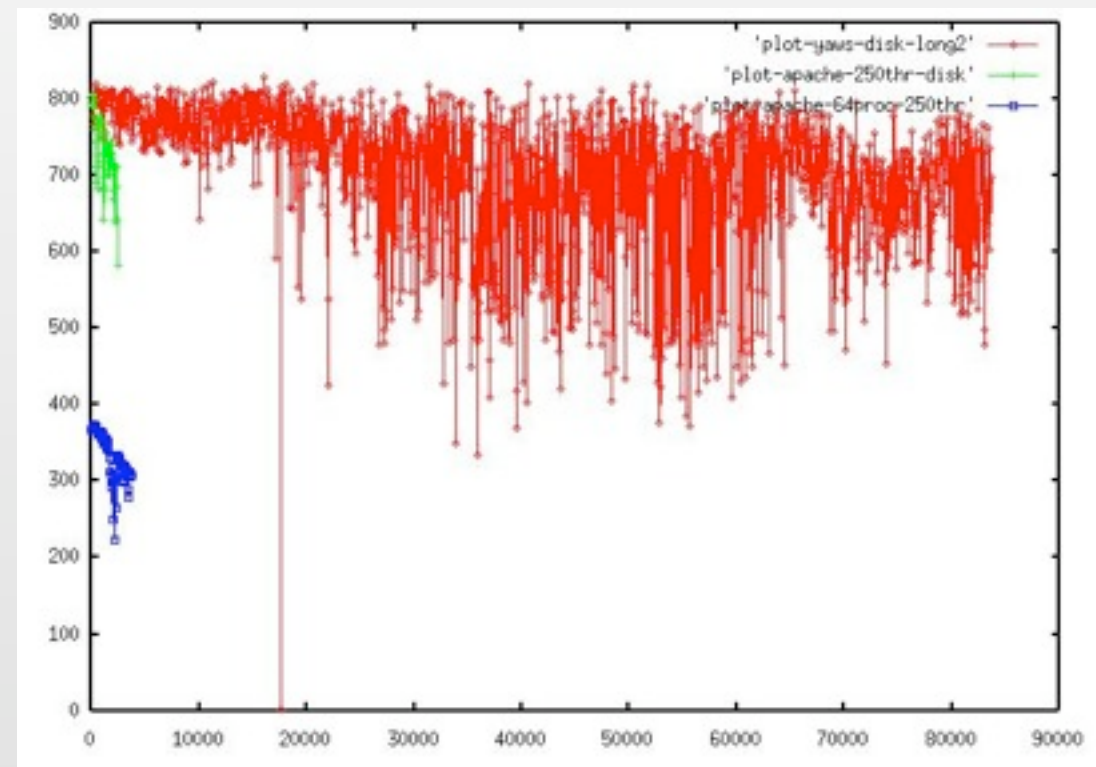
Erlang

Parallel

Fault tolerant

Addictive

Ninja Syntax



<http://www.sics.se/~joe/apachevsyaws.html>

BrowserCouch Test Suite

This file contains a simple test runner and suite for BrowserCouch.

The suite can be run [here](#).

The Runner

The `Testing` namespace is a simple test framework that supports the skipping of tests (for when the host system doesn't support required functionality for a test's execution), as well as asynchronous tests.

It doesn't currently have the ability to detect when an asynchronous test has failed, however.

Browser Couch Tests

You can read the source code documentation for these tests [here](#).

```
testDictionary OK
testViewMap_async OK
testViewMapFindRow_async OK
testViewProgress_async OK
testViewMapReduceFindRow_async OK
testViewMapReduceWebWorker_async running
testViewMapReduce_async pending
testLocalStorage_async pending
```

```
testDictionary OK
testViewMap_async OK
testViewMapFindRow_async OK
testViewProgress_async OK
testViewMapReduceFindRow_async OK
testViewMapReduceWebWorker_async running
testViewMapReduce_async pending
testLocalStorage_async pending
```

The Suite

The `Tests` namespace contains the actual testing suite for BrowserCouch.

Browser Couch

JavaScript port

Uses HTML5 storage

Replicates with CouchDB

<http://hg.toolness.com/browser-couch/>

```
var Testing = {
  run: function(listener, container, setTimeout) {
    if (!setTimeout)
      setTimeout = window.setTimeout;

    var tests = [];

    for (name in container)
      if (name.indexOf("test") == "0") {
        var test = {
          name: name,
          func: container[name],
          isAsync: name.indexOf("_async") != -1,
          id: tests.length,
          assertEquals: function assertEquals(a, b) {
            if (a != b)
              throw new Error(a + " != " + b);
          }
        };
        tests.push(test);
      }

    listener.onReady(tests);
    var nextTest = 0;

    function runNextTest() {
      if (nextTest < tests.length) {
        var test = tests[nextTest];
        listener.onRun(test);
        test.skip = function() {
          listener.onSkip(this);
          setTimeout(runNextTest, 0);
        };
        test.done = function() {
          listener.onFinish(this);
          setTimeout(runNextTest, 0);
        };
        test.func.call(container, test);
        if (!test.isAsync)
          test.done();
        nextTest++;
      }
    }

    runNextTest();
  }
};

var Tests = {
  testDictionary: function(self) {
    var dict = new BrowserCouch._Dictionary();
    dict.set('foo', {a: 'hello'});
    dict.set('bar', {b: 'goodbye'});
    self.assertEqual(dict.get('foo').a, 'hello');
    self.assertEqual(dict.get('bar').b, 'goodbye');
    self.assertEqual(dict.getKeys().length, 2);
    self.assertEqual(dict.has('foo'), true);
    self.assertEqual(dict.has('bar'), true);
    self.assertEqual(dict.has('spatula'), false);
    dict.remove('bar');
    self.assertEqual(dict.getKeys().length, 1);
    self.assertEqual(dict.has('foo'), true);
  },
  _setupTestDb: function(cb) {
    var documents = this._testDbContents;
    BrowserCouch.get(
      "blarg",
      function(db) {
        db.wipe(
          function() {
            db.put(
              documents,
              function() {
                ModuleLoader.require(
                  "JSON",
                  function() { cb(db); }
                );
              }
            );
          }
        );
      }
    );
    new FakeStorage()
  },
  _testDbContents: [{id: "monkey",
    content: "hello there dude"},
    {id: "chuck",
    content: "hello there dude"}]
```

No SQL

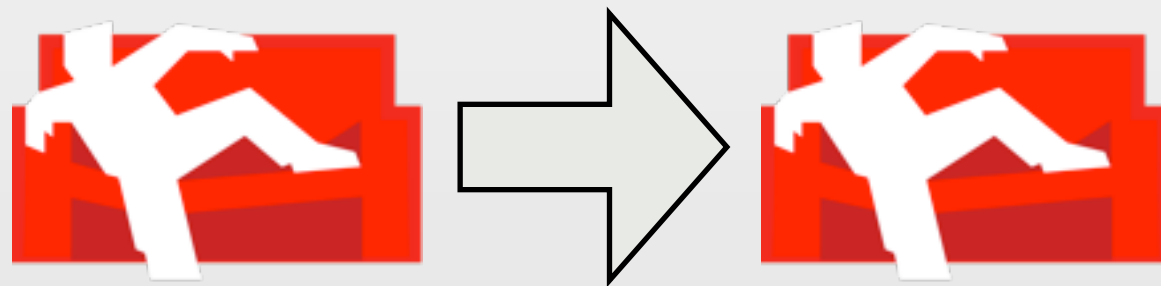
in

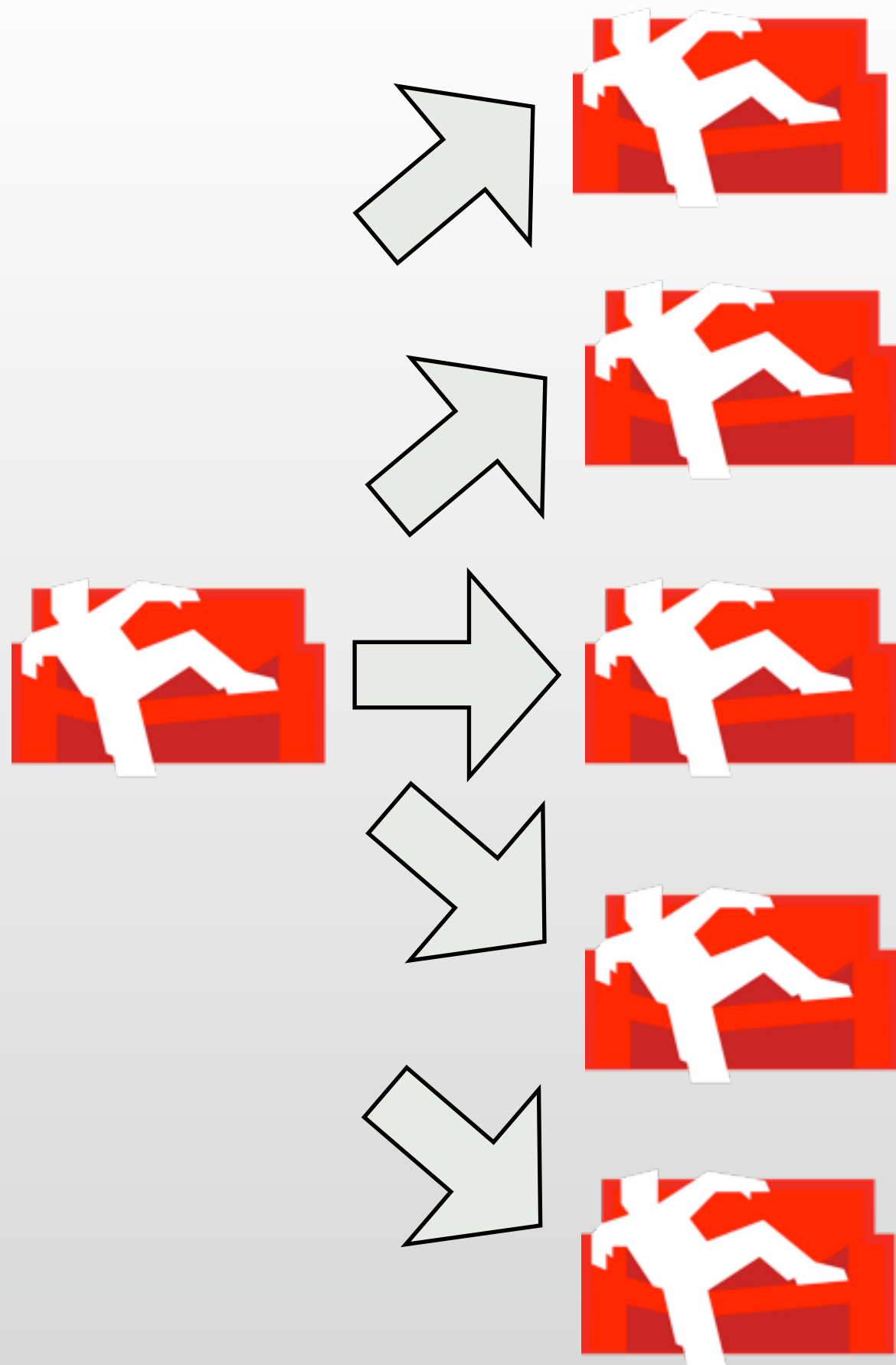
HTML5

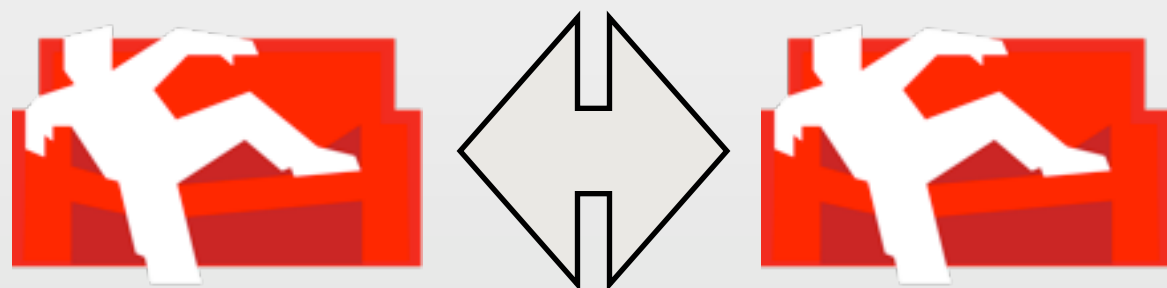
Incremental Replication

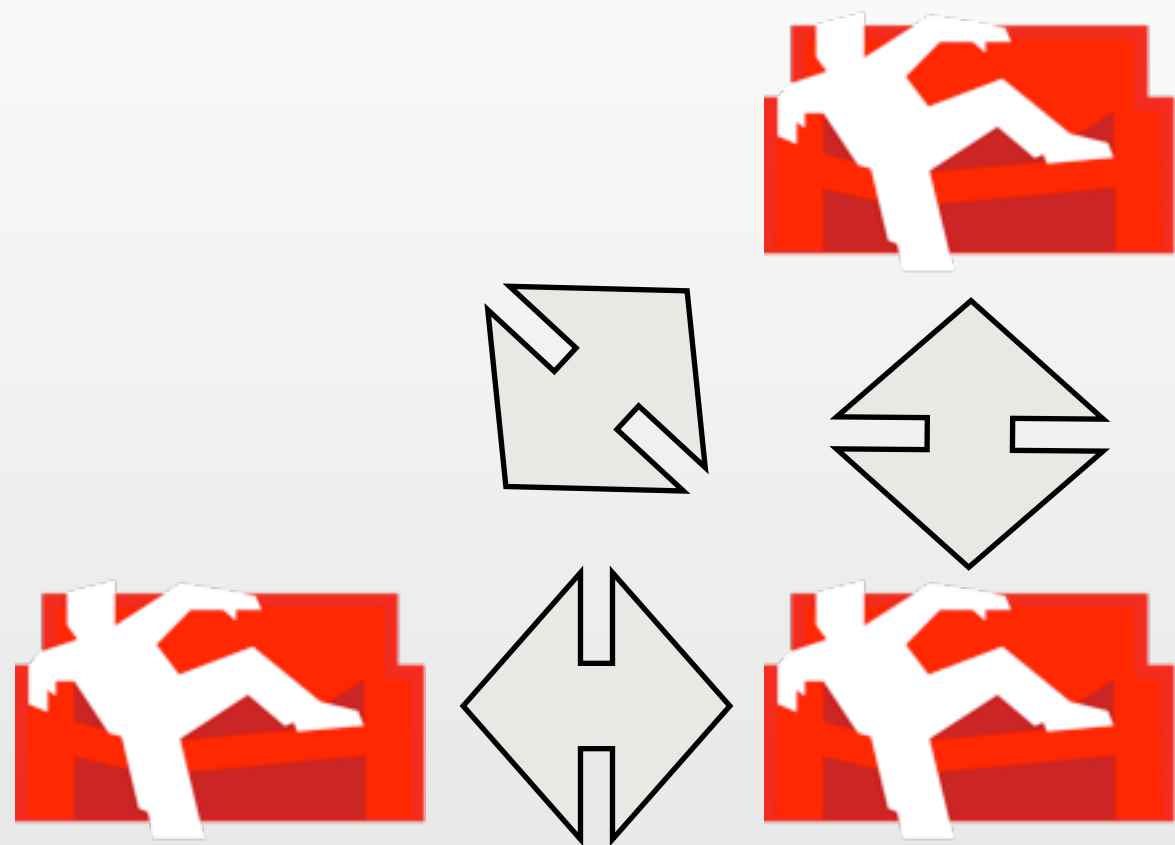


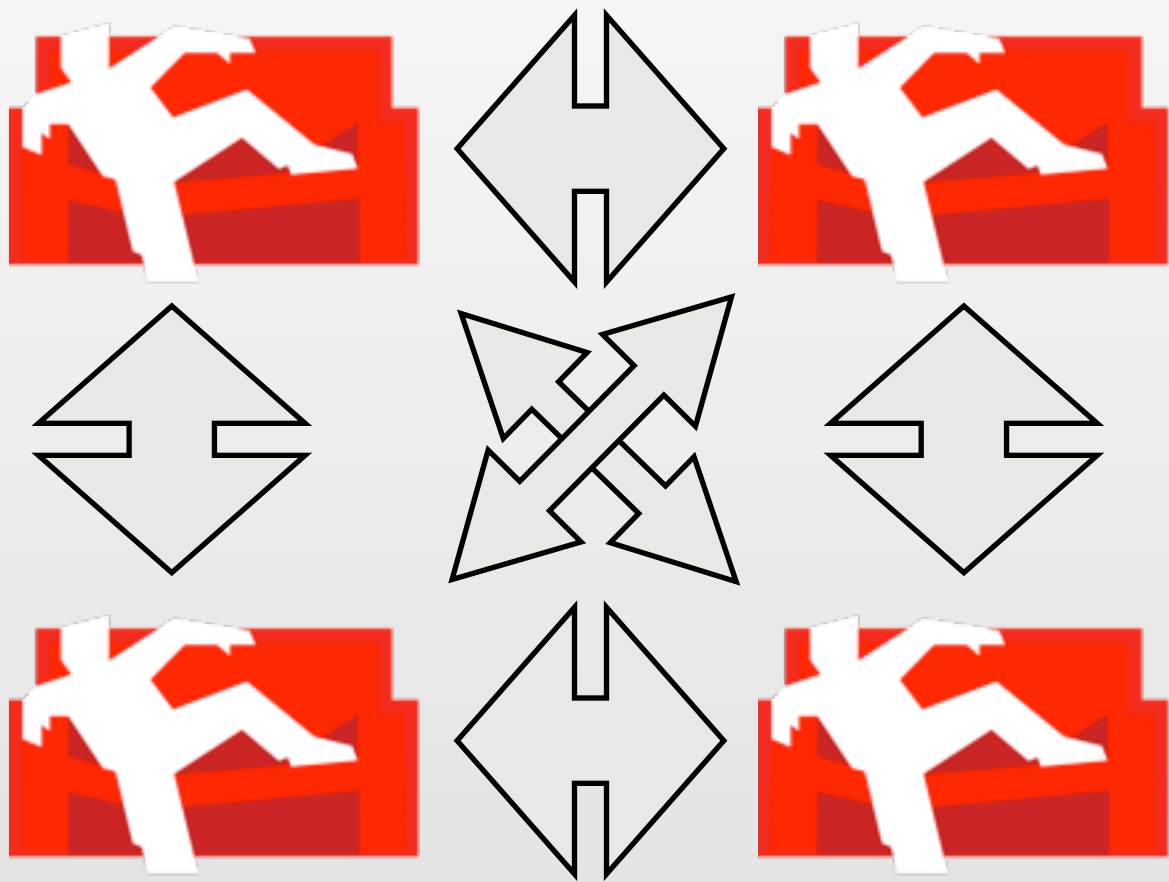
```
curl -X POST http://127.0.0.1:5984/_replicate -d  
'{"source": "http://couch.example.com:5984/remote-db",  
  "target": "local-db"}'
```

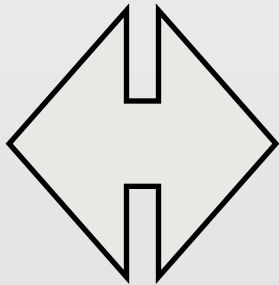
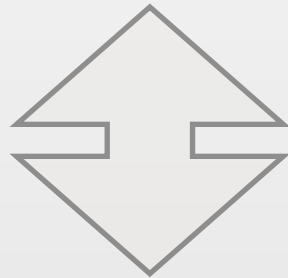
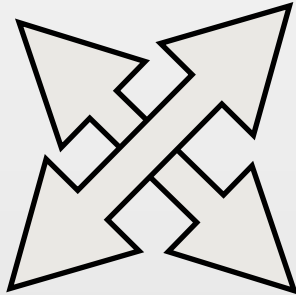
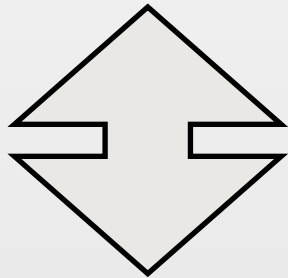


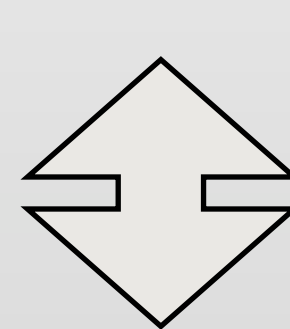
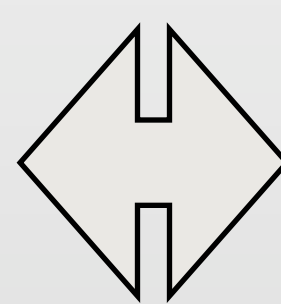
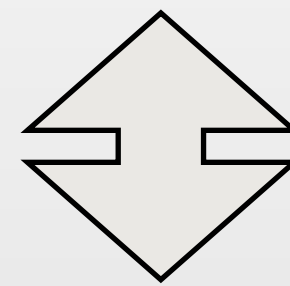
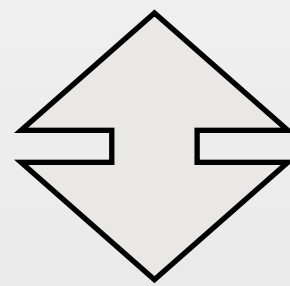
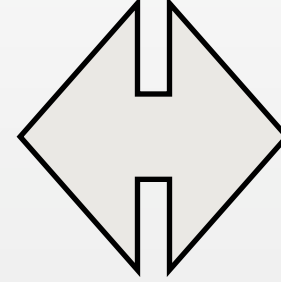
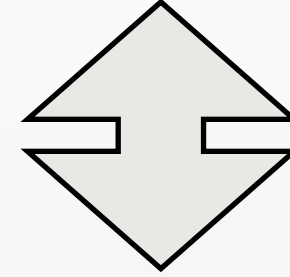


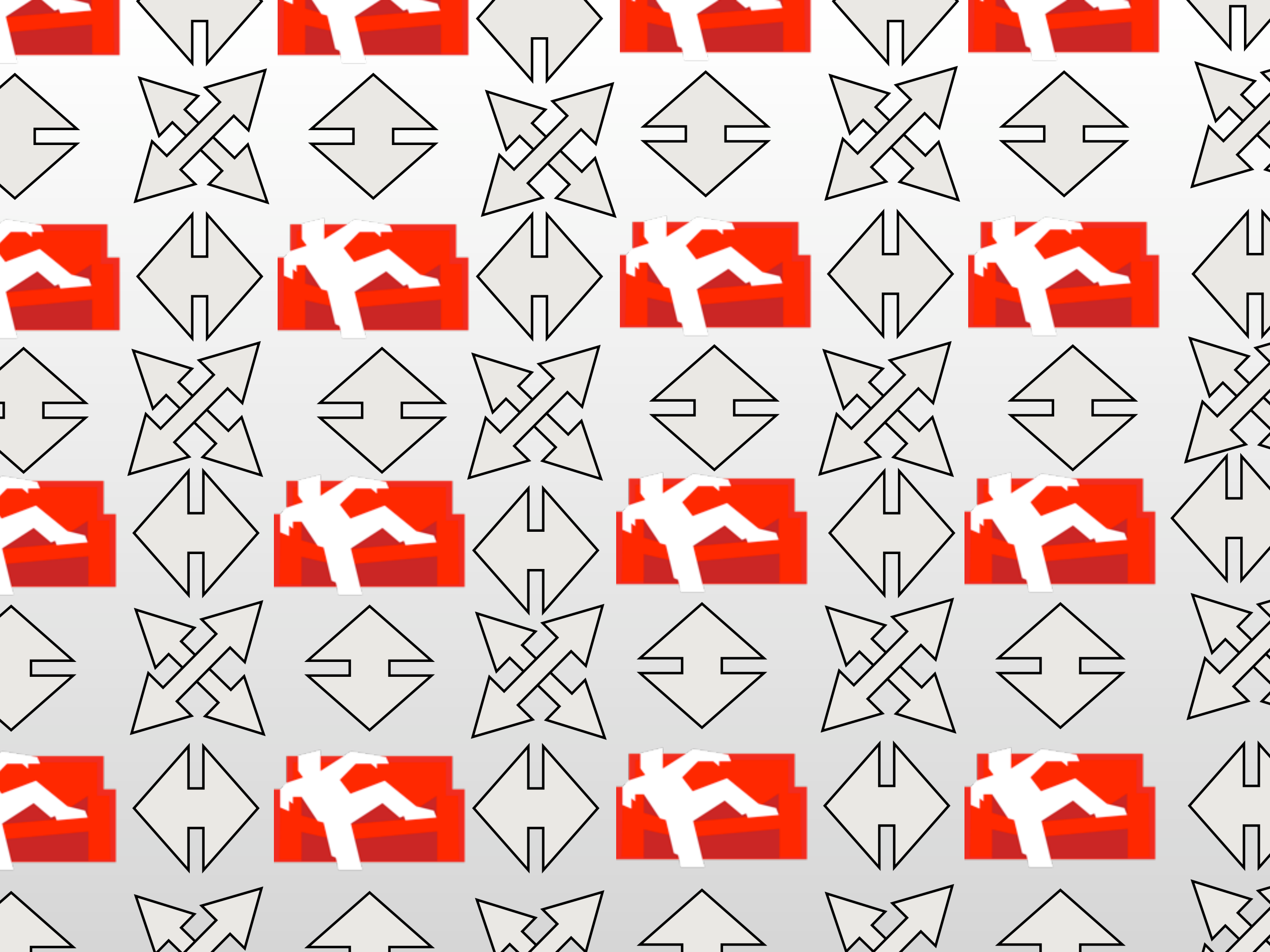












COUCH DB

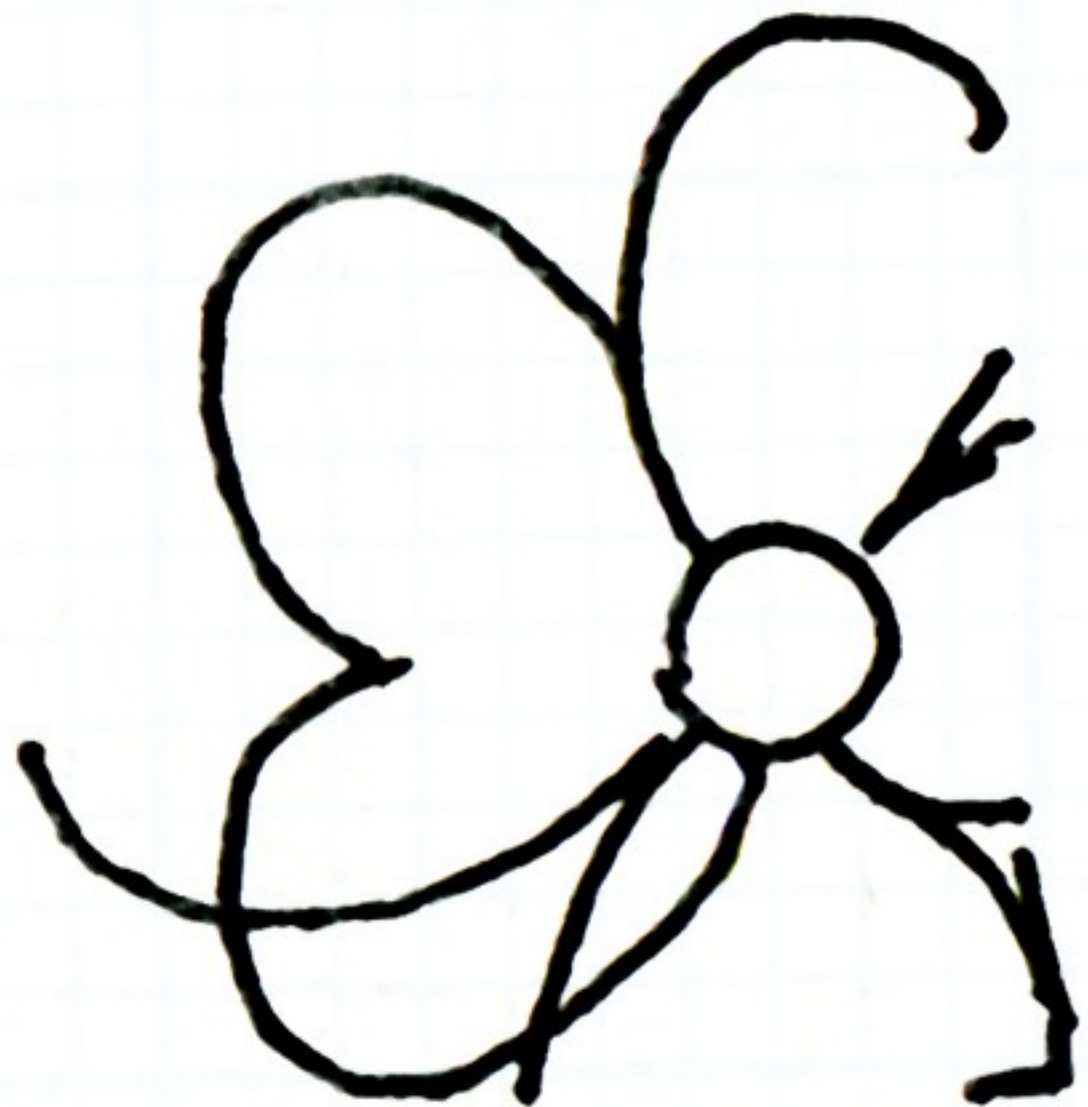
to
the

EDGE

Programming

at the

Edge



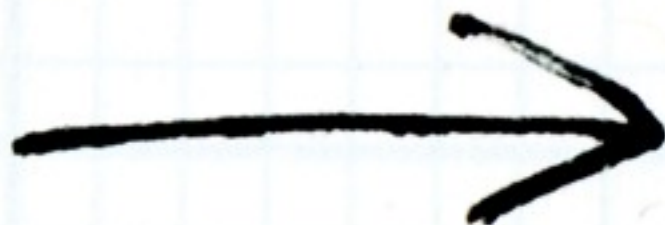
Context is
Expensive



Document Modeling

{ "_id": "foo",

"_rev": "2..." }



20

and design /

Documents

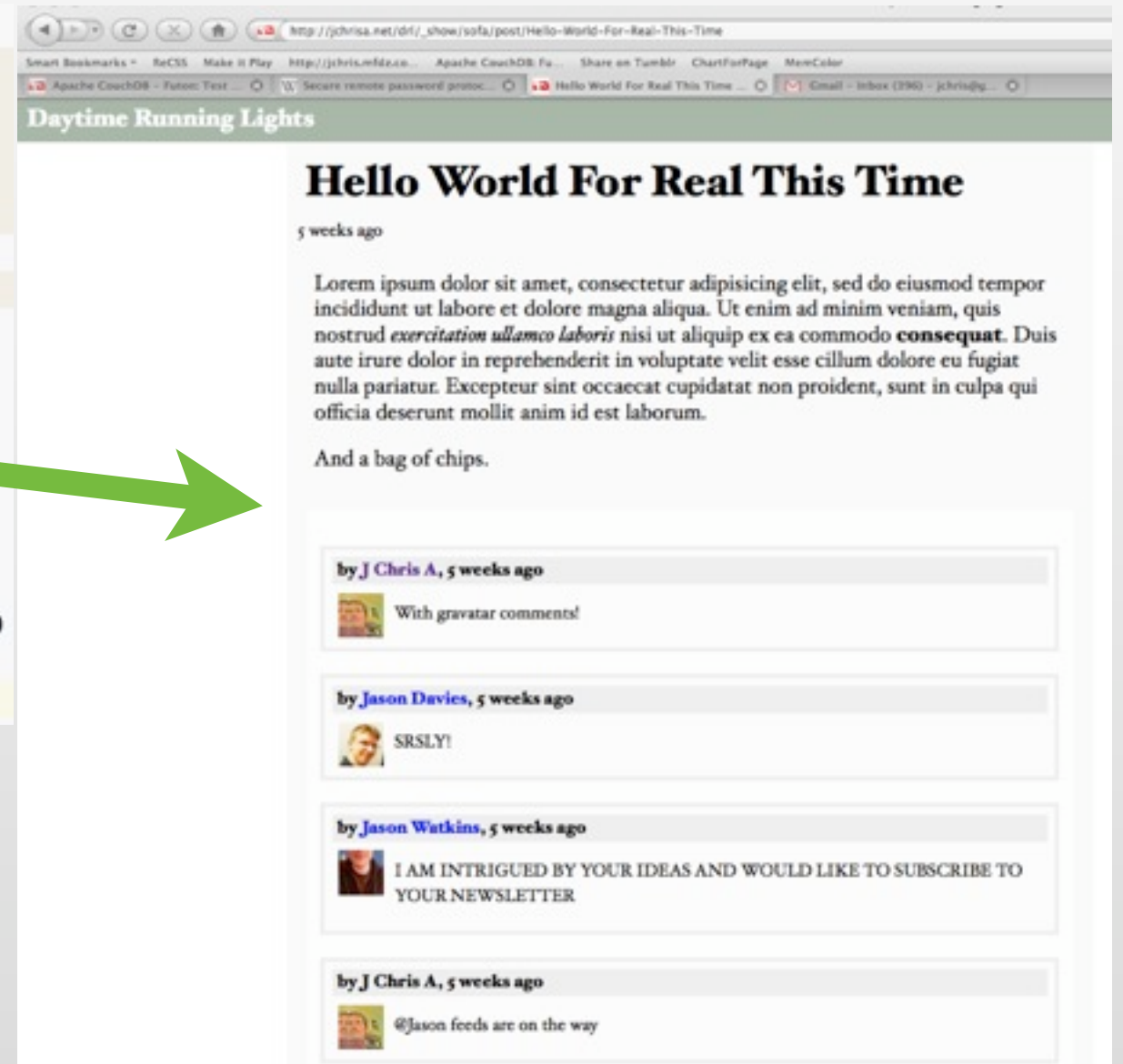
- map/reduce
- validations
- render html

Render JSON Docs as HTML

shows/post.js

/drl/_design/sofa/_show/post/Hello-World-For-Real-This-Time

```
function(doc, req) {  
  // !json templates.post  
  // !json blog  
  // !code helpers.template  
  // !code helpers.couchapp  
  // log(req.headers.Accept);  
  
  // we only show html  
  return template(templates.post, {  
    title : doc.title,  
    blogName : blog.title,  
    post : doc.html,  
    date : doc.created_at,  
    author : doc.author,  
    assets : assetPath(),  
    editPostPath : showPath('edit', doc._id),  
    index : listPath('index', 'recent-posts', {descending:true, limit:8})  
  });  
}
```



┌ Show 1 Doc ┐

- no side effects
- cacheable

└ ┘

Render Views as HTML

lists/index.js

/drl/_list/sofa/index/recent-posts?descending=true&limit=8

```
1 function(head, row, req) {
2   // json templates.index
3   // json blog
4   // code helpers.couchapp
5   // code helpers.template
6   // log(req.headers.Accept);
7   var indexPath = listPath('index', 'recent-posts', {descending:true, limit:8});
8   var feedPath = listPath('index', 'recent-posts', {descending:true, limit:8, format:'atom'});
9   return respondWith(req, {
10    html : function() {
11      if (head) {
12        return template(templates.index.head, {
13          title : blog.title,
14          newPostPath : showPath('edit'),
15          index : indexPath,
16          assets : assetPath()
17        });
18      } else if (row) {
19        var post = row.value;
20        return template(templates.index.row, {
21          title : post.title,
22          summary : post.summary,
23          date : post.created_at,
24          link : showPath('post', row.id),
25          assets : assetPath()
26        });
27      } else {
28        return template(templates.index.tail, {
29          assets : assetPath()
30        });
31      }
32    },
33    atom : function() {
34      // with first row in head you can do updated.
35      if (head) {
36        var f = <feed xmlns="http://www.w3.org/2005/Atom">;
37        f.title = blog.title;
38        f.id = makeAbsolute(req, indexPath);
39        f.link.href = makeAbsolute(req, feedPath);
40        f.link.rel = "self";
41        f.generator = "Sofa on CouchDB";
42        f.updated = new Date().rfc3339();
43        return (body:f.toString().replace(/\<\>/g, '&lt;'));
44      } else if (row) {
45        var entry = <entry>;
46        entry.id = makeAbsolute(req, '/' + encodeURIComponent(req.info.db_name) + '/' + encodeURIComponent(row.value.id));
47        entry.title = row.value.title;
48        entry.content = row.value.summary;
49        entry.content.type = "html";
50        entry.updated = new Date(row.value.created_at).rfc3339();
51        entry.author = <author><name>{row.value.author}</name></author>;
52        entry.link.href = makeAbsolute(req, showPath('post', row.id));
53        entry.link.rel = "alternate";
54        return (body:entry);
55      } else {
56        return (body : "</feed>");
57      }
58    }
59  });
60 }
```

Daytime Running Lights

web, vinyl, Scala, quote, photo,
music, lorem, life, hello world,
dev, couchdb, Couch, code couchapp,
code

Recently...

Peer Commit
7 days ago
I don't know the literature, but I have an idea for a shared data-space across peer nodes (not in a cluster). A replication helper, which tracks all known replicas, and their last successfully replicated sequence num. In this way, Node A can, for each update, provide to the user a list of remote nodes that have knowledge of the update. I think t...

Old News from a New Framework
12 days ago
Sling is a Scala app server for building Ajax CouchDB apps. The second half of this quote is Old News: By taking advantage of these characteristics pure-CouchDB applications can reach near parity with traditional server-heavy approaches. Notable exceptions include the ability to render dynamic text that isnâ€™t JSON, or return results that requi...

Deployed Sofa
16 days ago
For more information about this blog software, visit the Sofa source code on Github I plan to run my blog from this software, and I've got a few other projects in the pipeline. Also, don't forget about the Twitter client. Same great software, great new address....

Talk Announcements
4 weeks ago
My conference calendar is rapidly filling up with exciting CouchDB things. Here's the full list of them, but also: A new episode of the CouchDB podcast is out. Download the mp3 or subscribe to the RSS feed CouchDB Talks Feb 25 in Portland. Demoing CouchApps at PDX.js (7pm) The Portland JavaScript Admirers Group had its first meetin...

More Lorem
5 weeks ago
Lorem Lorem Lorem ...

[List 1 Row]

at a time

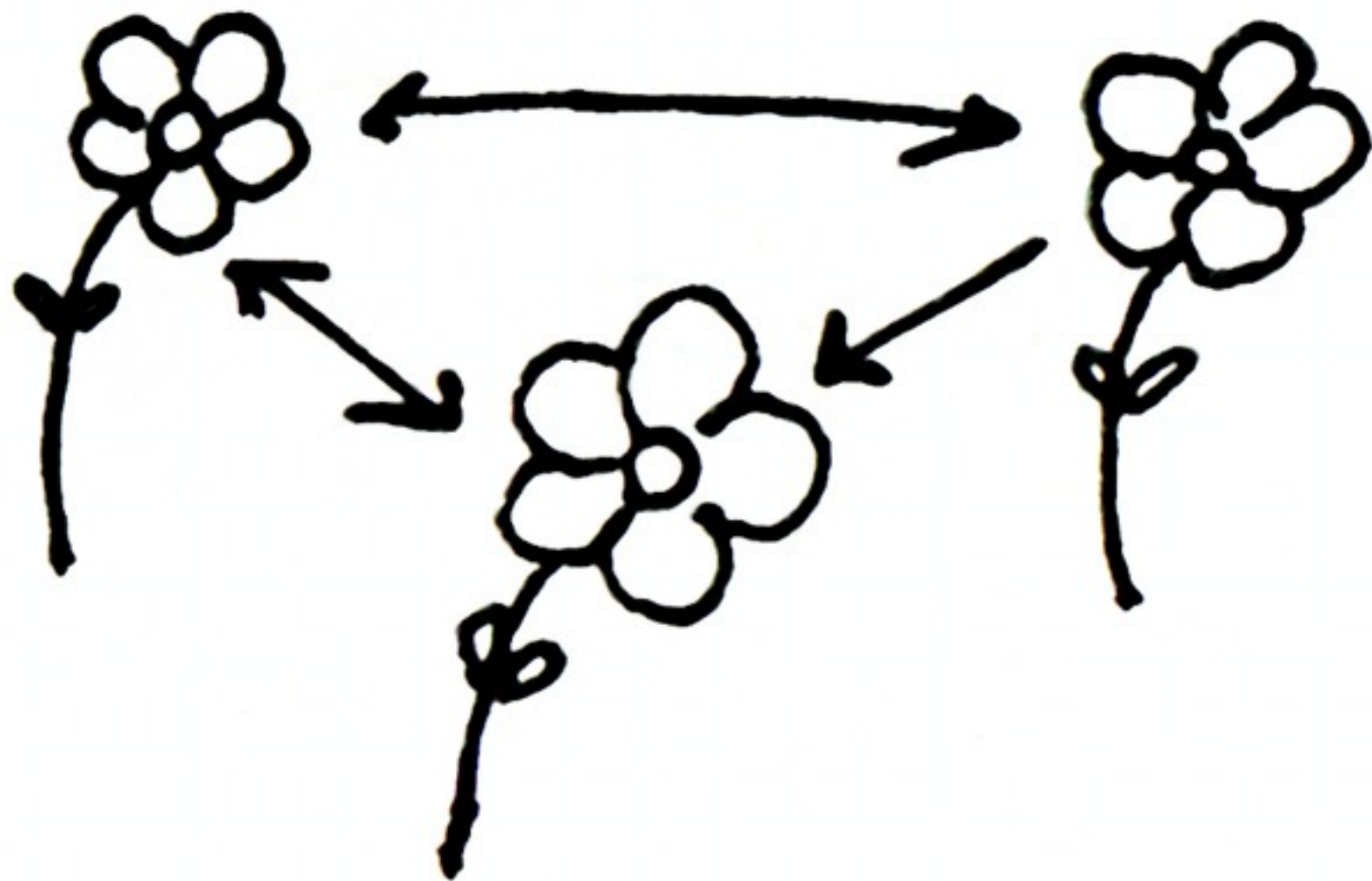
- easy not to
leak memory

Validate

1

Update

Replication



"normal"



server

offline

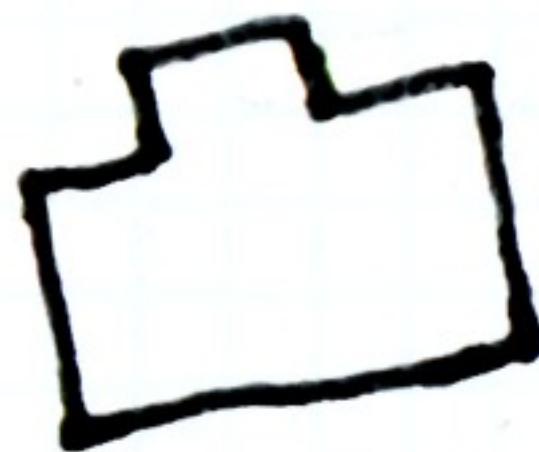
mode

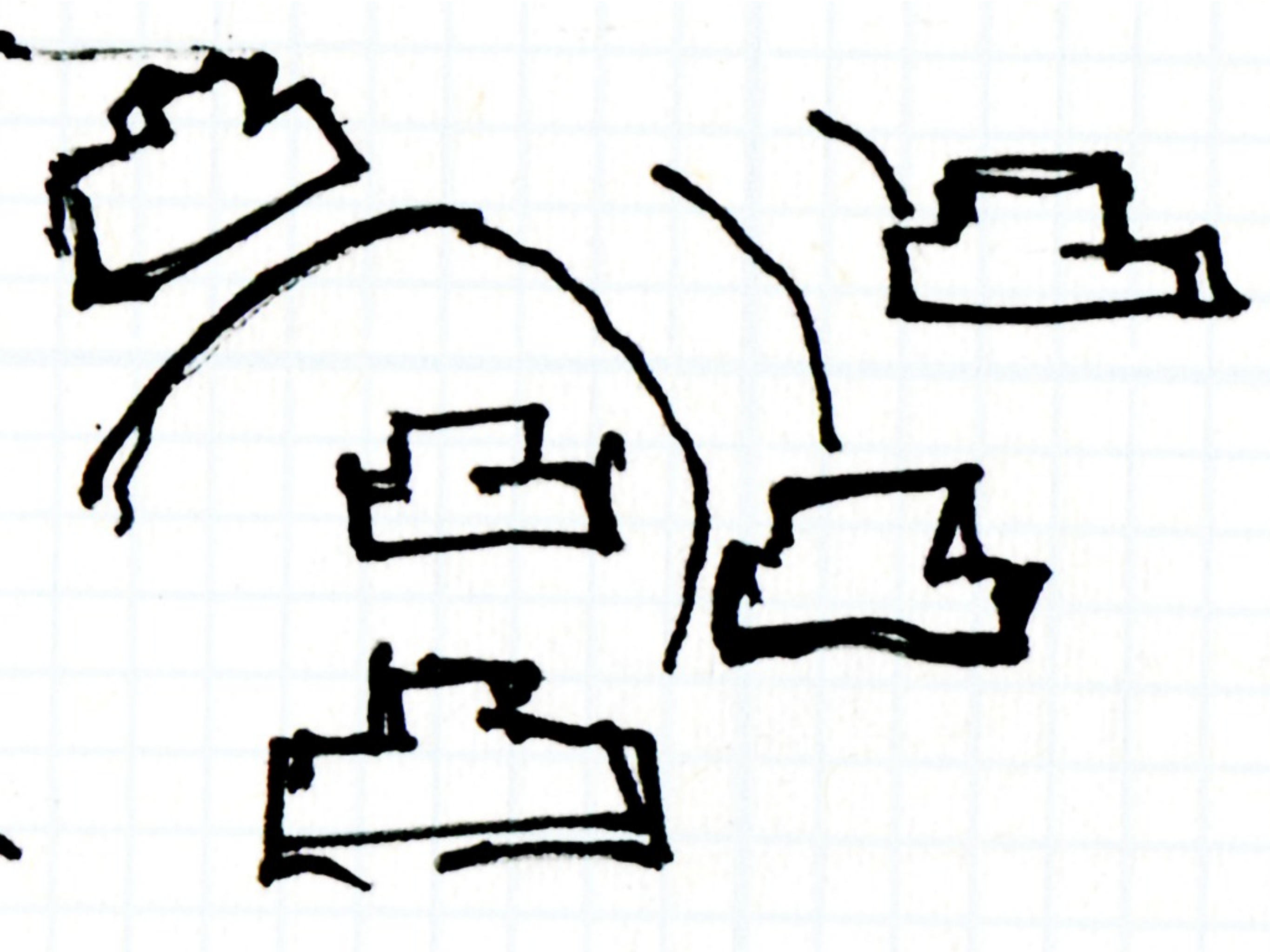


the p2p



web.





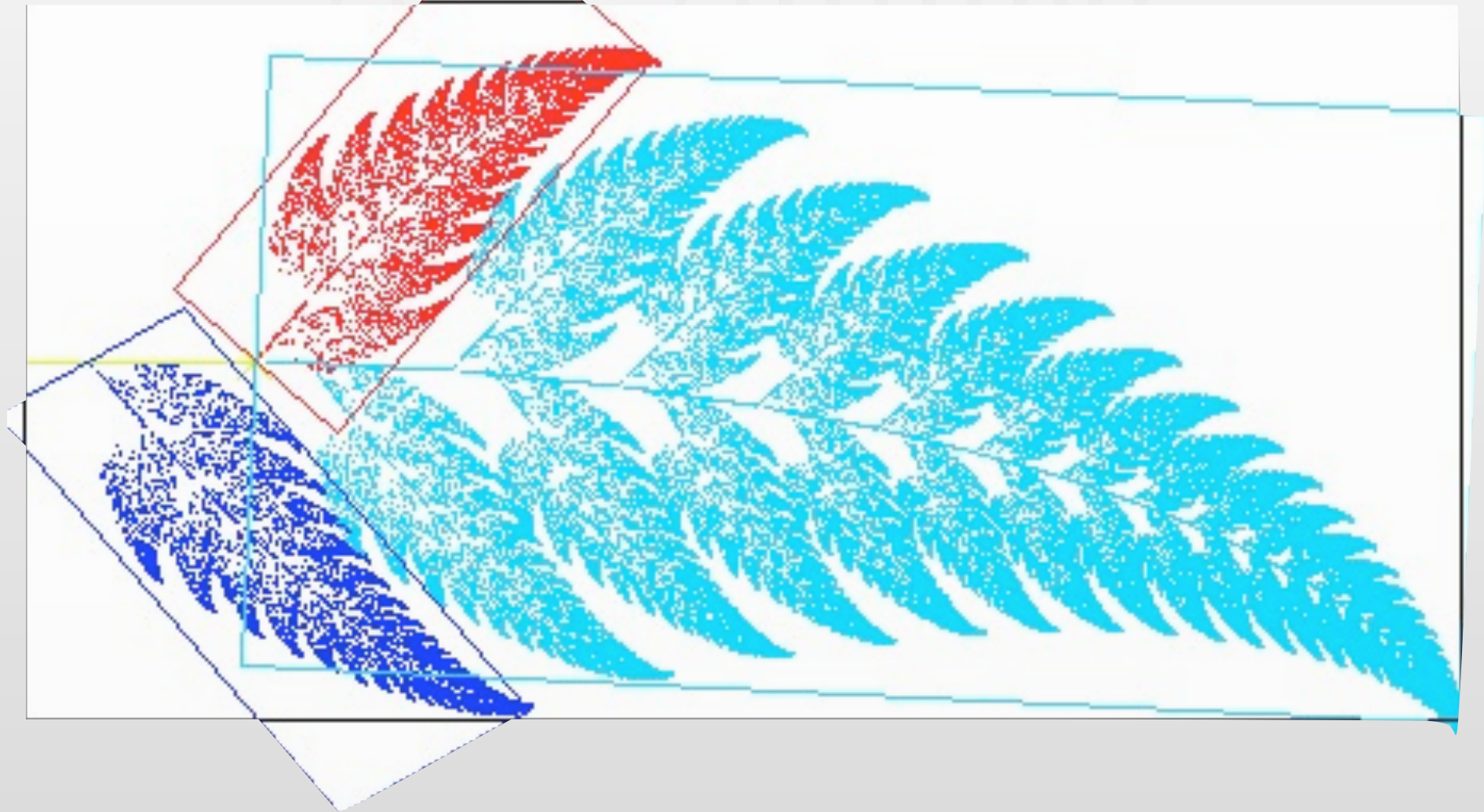
Identity



- public key
- signed updates

Self-Similar

Self-Similar
Self-Similar
Self-Similar



couch_btree.erl

Serialized Writes

Parallel Reads

$O(\log N)$

Consistent Snapshot

Append Only

Replication

`_all_docs_by_seq`

Idempotent / Resilient

Incremental

Compaction

by_seq

Generational garbage
collection

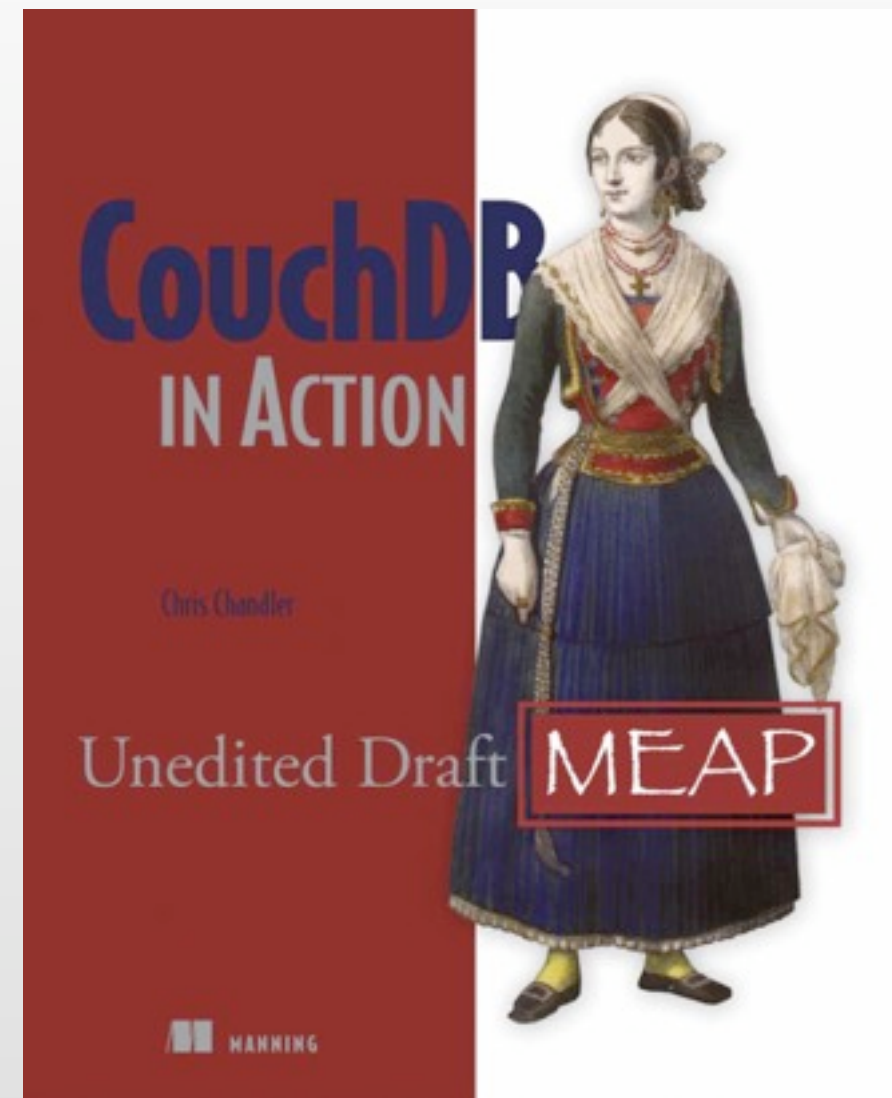
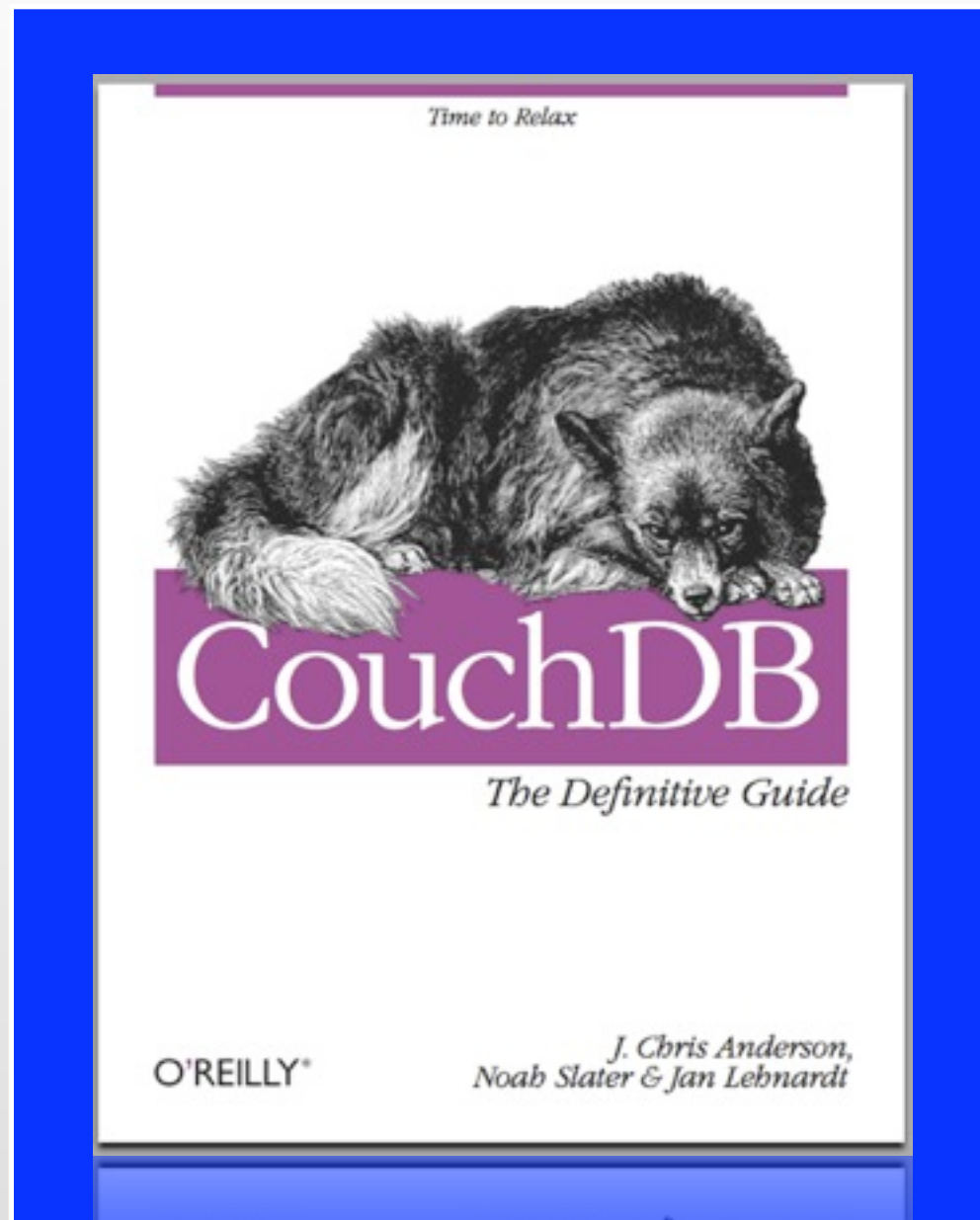
Views

Incremental

Intermediate reductions
stored in B-Tree inner nodes



@couchdbinaction



<http://books.couchdb.org/relax>

Resources

@CouchDB

<http://couchdb.apache.org/>

Dress like a Couch: <http://shop.couchdb.com>

<http://planet.couchdb.org/>

<https://peepcode.com/products/couchdb-with-rails>

couch.io

Berlin – London – Portland

Thanks!